

PYTHON PROGRAMMING LABORATORY MANUAL

(R201250)

I/IV B.Tech, Semester-I

Academic Year: 2021-22



FIRST YEAR ENGINEERING DEPARTMENT

Prepared by: T N S KOTI MANI KUMAR

CH SUDHAKAR

Y ANJANI

SIR C R REDDY COLLEGE OF ENGINEERING

Eluru-534007, West Godavari Dist, Andhra Pradesh, India

(Approved by AICTE, New Delhi & Permanently affiliated to JNTUK, Kakinada)

Telephone No: 08812-230840, 230565, Fax: 08812-224193

Website: www.sircrrengg.ac.in



**SIR C.R.REDDY COLLEGE OF ENGINEERING
ELURU-534007, WEST GODAVARI DIST, A P., INDIA**

(Approved by AICTE, New Delhi)

Phone no: 08812-230840, 2300656 Fax: 08812-224193

Visit us at <http://www.sircrrengg.ac.in>


FIRST YEAR ENGINEERING DEPARTMENT

VISION OF THE DEPARTMENT

Endeavour to make the department soar towards excellence in technical education by laying a firm foundation in academic pursuits for enhancement of personal qualities.

MISSION OF THE DEPARTMENT

- To create a healthy learning ambience wherein students enjoy acquisition of knowledge.
- To instill discipline in students and make them technologically superior and ethically strong.
- To enable students realize their teeming potential and use it to the fullest under the guidance of dedicated faculty.

	<p align="center"> SIR C.R.REDDY COLLEGE OF ENGINEERING ELURU-534007, WEST GODAVARI DIST, A P., INDIA (Approved by AICTE, New Delhi) Phone no: 08812-230840, 2300656 Fax: 08812-224193 Visit us at http://www.sircrrengg.ac.in FIRST YEAR ENGINEERING DEPARTMENT </p>
---	--

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/Development of Solutions: Design solutions for complex engineering problems and system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, society, and environmental considerations.

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in society and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.


PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multi-disciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi-disciplinary environments.


PO12: Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

	<p align="center">SIR C.R.REDDY COLLEGE OF ENGINEERING ELURU-534007, WEST GODAVARI DIST, A P., INDIA (Approved by AICTE, New Delhi) Phone no: 08812-230840, 2300656 Fax: 08812-224193 Visit us at http://www.sircrrengg.ac.in FIRST YEAR ENGINEERING DEPARTMENT</p>
---	---

GENERAL LABORATORY INSTRUCTIONS

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.
3. Student should enter into the laboratory with:
 - a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session.
 - b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab.
 - c. Proper Dress code and Identity card.
4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab observation note book, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintain the discipline and decency in the laboratory.

7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out ; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the lab after completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

	<p align="center"> SIR C.R.REDDY COLLEGE OF ENGINEERING ELURU-534007, WEST GODAVARI DIST, A P., INDIA (Approved by AICTE, New Delhi) Phone no: 08812-230840, 2300656 Fax: 08812-224193 Visit us at http://www.sircrrengg.ac.in FIRST YEAR ENGINEERING DEPARTMENT </p>
---	--

SYLLABUS

List of Experiments:

- 1) Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram.
- 2) Write a program that asks the user to enter three numbers (use three separate input statements). Create variables called total and average that hold the sum and average of the three numbers and print out the values of total and average.
- 3) Write a program that uses a for loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86, 89.
- 4) Write a program that asks the user for their name and how many times to print it. The program should print out the user's name the specified number of times.
- 5) Use a for loop to print a triangle like the one below. Allow the user to specify how high the triangle should be.


```

*
**
***
****
      
```
- 6) Generate a random number between 1 and 10. Ask the user to guess the number and print a message based on whether they get it right or not.
- 7) Write a program that asks the user for two numbers and prints Close if the numbers are within .001 of each other and Not close otherwise.
- 8) Write a program that asks the user to enter a word and prints out whether that word contains any vowels.
- 9) Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the

characters of the two strings. For example, if the user enters abcde and ABCDE the program should print out AaBbCcDdEe.

10) Write a program that asks the user for a large integer and inserts commas into it according to the standard American convention for commas in large numbers. For instance, if the user enters 1000000, the output should be 1,000,000.

11) In algebraic expressions, the symbol for multiplication is often left out, as in $3x+4y$ or $3(x+5)$. Computers prefer those expressions to include the multiplication symbol, like $3*x+4*y$ or $3*(x+5)$. Write a program that asks the user for an algebraic expression and then inserts multiplication symbols where appropriate.

12) Write a program that generates a list of 20 random numbers between 1 and 100. (a) Print the list. (b) Print the average of the elements in the list. (c) Print the largest and smallest values in the list. (d) Print the second largest and second smallest entries in the list (e) Print how many even numbers are in the list.

13) Write a program that asks the user for an integer and creates a list that consists of the factors of that integer.

14) Write a program that generates 100 random integers that are either 0 or 1. Then find the longest run of zeros, the largest number of zeros in a row. For instance, the longest run of zeros in $[1,0,1,1,0,0,0,1,0,0]$ is 4.

15) Write a program that removes any repeated items from a list so that each item appears at most once. For instance, the list $[1,1,2,3,4,3,0,0]$ would become $[1,2,3,4,0]$.

16) Write a program that asks the user to enter a length in feet. The program should then give the user the option to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometers. Say if the user enters a 1, then the program converts to inches, if they enter a 2, then the program converts to yards, etc. While this can be done with if statements, it is much shorter with lists and it is also easier to add new conversions if you use lists.

17) Write a function called `sum_digits` that is given an integer `num` and returns the sum of the digits of `num`.

18) Write a function called `first_diff` that is given two strings and returns the first location in which the strings differ. If the strings are identical, it should return -1.

19) Write a function called `number_of_factors` that takes an integer and returns how many factors the number has.

- 20) Write a function called `is_sorted` that is given a list and returns `True` if the list is sorted and `False` otherwise.
- 21) Write a function called `root` that is given a number `x` and an integer `n` and returns $x^{1/n}$. In the function definition, set the default value of `n` to 2.
- 22) Write a function called `primes` that is given a number `n` and returns a list of the first `n` primes. Let the default value of `n` be 100.
- 23) Write a function called `merge` that takes two already sorted lists of possibly different lengths, and merges them into a single sorted list. (a) Do this using the `sort` method. (b) Do this without using the `sort` method.
- 24) Write a program that asks the user for a word and finds all the smaller words that can be made from the letters of that word. The number of occurrences of a letter in a smaller word can't exceed the number of occurrences of the letter in the user's word.
- 25) Write a program that reads a file consisting of email addresses, each on its own line. Your program should print out a string consisting of those email addresses separated by semicolons.
- 26) Write a program that reads a list of temperatures from a file called `temps.txt`, converts those temperatures to Fahrenheit, and writes the results to a file called `ftemps.txt`.
- 27) Write a class called `Product`. The class should have fields called `name`, `amount`, and `price`, holding the product's name, the number of items of that product in stock, and the regular price of the product. There should be a method `get_price` that receives the number of items to be bought and returns a the cost of buying that many items, where the regular price is charged for orders of less than 10 items, a 10% discount is applied for orders of between 10 and 99 items, and a 20% discount is applied for orders of 100 or more items. There should also be a method called `make_purchase` that receives the number of items to be bought and decreases `amount` by that much.
- 28) Write a class called `Time` whose only field is a time in seconds. It should have a method called `convert_to_minutes` that returns a string of minutes and seconds formatted as in the following example: if `seconds` is 230, the method should return `'5:50'`. It should also have a method called `convert_to_hours` that returns a string of hours, minutes, and seconds formatted analogously to the previous method.
- 29) Write a class called `Converter`. The user will pass a length and a unit when declaring an object from the class—for example, `c = Converter(9,'inches')`. The possible units are inches, feet, yards, miles, kilometers, meters, centimeters, and millimeters. For each of these units there should be a method that returns the length converted into those units. For example, using the `Converter` object created above, the user could call `c.feet()` and should get 0.75 as the result.

- 30) Write a Python class to implement $\text{pow}(x, n)$.
- 31) Write a Python class to reverse a string word by word.
- 32) Write a program that opens a file dialog that allows you to select a text file. The program then displays the contents of the file in a textbox.
- 33) Write a program to demonstrate Try/except/else.
- 34) Write a program to demonstrate try/finally and with/as.

	<p align="center"> SIR C.R.REDDY COLLEGE OF ENGINEERING ELURU-534007, WEST GODAVARI DIST, A P., INDIA (Approved by AICTE, New Delhi) Phone no: 08812-230840, 2300656 Fax: 08812-224193 Visit us at http://www.sircrrengg.ac.in FIRST YEAR ENGINEERING DEPARTMENT </p>
---	--

COURSE OUTCOMES

CO	OUTCOMES
1	Understand the need for learning basic concepts of Python programming language
2	Apply various data structures in developing solutions to real time scenarios
3	Analyze various concepts of functions, make use of packages, FILES, object oriented concepts in python programming and Outline Exception handling concepts.



SIR C.R.REDDY COLLEGE OF ENGINEERING
ELURU-534007, WEST GODAVARI DIST, A P., INDIA
 (Approved by AICTE, New Delhi)
 Phone no: 08812-230840, 2300656 Fax: 08812-224193
 Visit us at <http://www.sircrrengg.ac.in>
FIRST YEAR ENGINEERING DEPARTMENT


INDEX

S. No	Name of Experiment	CO	Page No
1	Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram.	1	15
2	Write a program that asks the user to enter three numbers (use three separate input statements). Create variables called total and average that hold the sum and average of the three numbers and print out the values of total and average	1	15
3	Write a program that uses a for loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86, 89.	1	16
4	Write a program that asks the user for their name and how many times to print it. The program should print out the user's name the specified number of times	1	16
5	Use a for loop to print a triangle like the one below. Allow the user to specify how high the triangle should be. * ** *** ****	1	17
6	Generate a random number between 1 and 10. Ask the user to guess the number and print a message based on whether they get it right or not.	1	18
7	Write a program that asks the user for two numbers and prints Close if the numbers are within .001 of each other and Not close otherwise.	1	19
8	Write a program that asks the user to enter a word and prints out whether that word contains any vowels.	1	19-20
9	Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If	1	20

	they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters abcde and ABCDE the program should print out AaBbCcDdEe		
10	Write a program that asks the user for a large integer and inserts commas into it according to the standard American convention for commas in large numbers. For instance, if the user enters 1000000, the output should be 1,000,000	1	21
11	In algebraic expressions, the symbol for multiplication is often left out, as in $3x+4y$ or $3(x+5)$. Computers prefer those expressions to include the multiplication symbol, like $3*x+4*y$ or $3*(x+5)$. Write a program that asks the user for an algebraic expression and then inserts multiplication symbols where appropriate.	1	21-22
12	Write a program that generates a list of 20 random numbers between 1 and 100. (a) Print the list. (b) Print the average of the elements in the list. (c) Print the largest and smallest values in the list. (d) Print the second largest and second smallest entries in the list (e) Print how many even numbers are in the list.	2	22-23
13	Write a program that asks the user for an integer and creates a list that consists of the factors of that integer	2	24
14	Write a program that generates 100 random integers that are either 0 or 1. Then find the longest run of zeros, the largest number of zeros in a row. For instance, the longest run of zeros in $[1,0,1,1,0,0,0,1,0,0]$ is 4.	3	25
15	Write a program that removes any repeated items from a list so that each item appears at most once. For instance, the list $[1,1,2,3,4,3,0,0]$ would become $[1,2,3,4,0]$.	2	25-26
16	Write a program that asks the user to enter a length in feet. The program should then give the user the option to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometers. Say if the user enters a 1, then the program converts to inches, if they enter a 2, then the program converts to yards, etc. While this can be done with if statements, it is much shorter with lists and it is also easier to add new conversions if you use lists.	2	27
17	Write a function called <code>sum_digits</code> that is given an integer <code>num</code> and returns the sum of the digits of <code>num</code>	1	28
18	Write a function called <code>first_diff</code> that is given two strings and returns the first location in which the strings differ. If the strings are identical, it should return -1.	2	29
19	Write a function called <code>number_of_factors</code> that takes an integer and returns how many factors the number has	3	30
20	Write a function called <code>is_sorted</code> that is given a list and returns True if the list is sorted and False otherwise	3	30
21	Write a function called <code>root</code> that is given a number <code>x</code> and an	3	31

	integer n and returns $x1/n$. In the function definition, set the default value of n to 2.		
22	Write a function called primes that is given a number n and returns a list of the first n primes. Let the default value of n be 100.	3	32
23	Write a function called merge that takes two already sorted lists of possibly different lengths, and merges them into a single sorted list. (a) Do this using the sort method. (b) Do this without using the sort method.	3	33
24	Write a program that asks the user for a word and finds all the smaller words that can be made from the letters of that word. The number of occurrences of a letter in a smaller word can't exceed the number of occurrences of the letter in the user's word	2	34
25	Write a program that reads a file consisting of email addresses, each on its own line. Your program should print out a string consisting of those email addresses separated by semicolons	3	35
26	Write a program that reads a list of temperatures from a file called temps.txt, converts those temperatures to Fahrenheit, and writes the results to a file called ftemps.txt.	3	36
27	Write a class called Product. The class should have fields called name, amount, and price, holding the product's name, the number of items of that product in stock, and the regular price of the product. There should be a method get_price that receives the number of items to be bought and returns a the cost of buying that many items, where the regular price is charged for orders of less than 10 items, a 10% discount is applied for orders of between 10 and 99 items, and a 20% discount is applied for orders of 100 or more items. There should also be a method called make_purchase that receives the number of items to be bought and decreases amount by that much.	3	37
28	Write a class called Time whose only field is a time in seconds. It should have a method called convert_to_minutes that returns a string of minutes and seconds formatted as in the following example: if seconds is 230, the method should return '5:50'. It should also have a method called convert_to_hours that returns a string of hours, minutes, and seconds formatted analogously to the previous method.	3	40
29	Write a class called Converter. The user will pass a length and a unit when declaring an object from the class—for example, <code>c = Converter(9,'inches')</code> . The possible units are inches, feet, yards, miles, kilometers, meters, centimeters, and millimeters. For each of these units there should be a method that returns the length converted into those units.	3	42

	For example, using the Converter object created above, the user could call c.feet() and should get 0.75 as the result		
30	Write a Python class to implement pow(x, n).	3	49
31	Write a Python class to reverse a string word by word.	3	50
32	Write a program that opens a file dialog that allows you to select a text file. The program then displays the contents of the file in a textbox	3	50
33	Write a program to demonstrate Try/except/else	3	51
34	Write a program to demonstrate try/finally and with/as.	3	52

	<p>SIR C.R.REDDY COLLEGE OF ENGINEERING ELURU-534007, WEST GODAVARI DIST, A P., INDIA (Approved by AICTE, New Delhi) Phone no: 08812-230840, 2300656 Fax: 08812-224193 Visit us at http://www.sircrrengg.ac.in FIRST YEAR ENGINEERING DEPARTMENT</p>
---	---

CONTENT BEYOND SYLLABUS

S. No	Name of Experiment	CO	Page No
1	Write a program that defines a matrix and prints	2	54
2	Write a program to perform addition of two square matrices	2	55

Program 1

Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram.

Program:

```
KG=float(input("Enter the weight in Kilograms:"))  
pounds=2.2*KG  
print("weight in pounds:",pounds)
```

Output:

```
Enter the weight in Kilograms:56  
weight in pounds: 123.20000000000002
```

Program 2

Write a program that asks the user to enter three numbers (use three separate input statements). Create variables called total and average that hold the sum and average of the three numbers and print out the values of total and average.

Program:

```
number1=int(input("enter the number1:"))  
number2=int(input("enter the number2:"))  
number3=int(input("enter the number3:"))  
Total=number1+number2+number3  
avg=Total/3  
print("Total=",Total)  
print("Average=",avg)
```

Output:

```
enter the number1:20
```

```
enter the number2:60
enter the number3:70
Total= 150
Average= 50.0
```

Program 3

Write a program that uses a for loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86, 89.

Program:

```
for i in range(8,90,3):
    print(i,end=" ")
```

Output:

```
8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53 56 59 62 65 68 71 74 77 80 83 86 89
```

Program 4

Write a program that asks the user for their name and how many times to print it. The program should print out the user's name the specified number of times.

Program:

```
name=input("Enter the Name:")
n=int(input("Enter how many Times you want to print:"))
for i in range(0,n):
    print(name)
```

Output:

```
Enter the Name:satya
Enter how many Times you want to print:5
satya
satya
satya
satya
satya
```


Program 5

Use a *for* loop to print a triangle like the one below. Allow the user to specify how high the triangle should be.

```
*  
**  
***  
****
```

Program:

```
height=int(input("Enter the height of the Triangle:"))  
  
for i in range(1,height+1):  
  
    for j in range(i):  
  
        print("*",end="")  
  
    print("")
```

Output:

```
Enter the height of the Triangle:6  
*  
**  
***  
****  
*****  
*****
```

Program-6

Generate a random number between 1 and 10. Ask the user to guess the number and print a message based on whether they get it right or not.

Program:

```
import random n=random.randint(1,10)

Guessnumber=int(input("Enter a Number Between 1 to 10:"))

print("Random Number is=",n)

if(n==Guessnumber):

    print("Your Guess is Right")

else:

    print("Your Guess is Wrong")
```

Output:

```
Enter a Number Between 1 to 10:5
Random Number is= 4
Your Guess is Wrong
```

```
Enter a Number Between 1 to 10:8
Random Number is= 5
Your Guess is Wrong
```

Program-7

Write a program that asks the user for two numbers and prints Close if the numbers are within .001 of each other and Not close otherwise.

Note: 'abs' is a **built-in math function** in python which is used for **absolute value**.
from decimal import *

Program:

```
num1 = Decimal(input("Enter number 1 : "))
num2 = Decimal(input("Enter number 2 : "))
diff=abs(num1-num2)
if (diff<=0.001):
    print("close")
else:
    print("Not Close")
```

Output:

```
Enter number 1 : 12.003
Enter number 2 : 12.004
close
```

```
Enter number 1 : 12.004
Enter number 2 : 12.0000456
Not Close
```

Program-8

Write a program that asks the user to enter a word and prints out whether that word contains any vowels.

Program:

```
word=input("Enter the word:")
count=0
for i in word:
    if(i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or
       i=='A' or i=='E' or i=='I' or i=='O' or i=='U'):
        count=count+1
        break;
```

```
if(count==1):
    print("Word contain vowel")
else:
    print("Word contain not vowel")
```

Output:

```
Enter the word: CRR
Word contain not vowel
```

```
Enter the word: CRR coe
Word contain vowel
```

Program-9

Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters abcde and ABCDE the program should print out AaBbCcDdEe.

Program:

```
s1=input("enter the string1:")
s2=input("enter the string2:")
r=""
if(len(s1)==len(s2)):
    print("Two strings are same length')
    for i in range(len(s1)):
        r=r+(s2[i]+s1[i])
    print(r)
else:
    print("Two strings are different")
```

Output:

```
enter the string1:abcd
```

```
enter the string2:ABCD
```

```
Two strings are same length
```

```
AaBbCcDd
```

```
enter the string1:cdf
```

enter the string2:ABCdf

Two strings are different

Program-10

Write a program that asks the user for a large integer and inserts commas into it according to the standard American convention for commas in large numbers. For instance, if the user enters 1000000, the output should be 1,000,000.

Program:

```
number=int(input("Enter the Long integer:"))  
print("{:,}".format(number))
```

Output:

Enter the Long integer:1000000

1,000,000

Enter the Long integer:33000

33,000

Program-11

In algebraic expressions, the symbol for multiplication is often left out, as in $3x+4y$ or $3(x+5)$. Computers prefer those expressions to include the multiplication symbol, like $3*x+4*y$ or $3*(x+5)$. Write a program that asks the user for an algebraic expression and then inserts multiplication symbols where appropriate.

```
s=input("Enter algebraic Expressions:")
```

```
l=list(s)
```

```
result=' '
```

```
i=0
```

```
while(i<len(l)):
```

```
    if l[i]=='(':
```

```

index=l.index(')')
s2=' '.join(l[i:index+1])
result=result+'*'+l[i]
i=i+len(s2)
else if l[i].isalpha():
    result=result+'*'+l[i]
    i=i+1
else
    result=result+l[i]
    i=i+1
print(result)

```

Output:

```

Enter algebraic expression:3x+4y
3*x+4*y

```

Program 12

Write a program that generates a list of 20 random numbers between 1 and 100.

- (a) Print the list.
- (b) Print the average of the elements in the list.
- (c) Print the largest and smallest values in the list.
- (d) Print the second largest and second smallest entries in the list
- (e) Print how many even numbers are in the list.

Program:

```
import random

l=[]

for num in range(20):

    l.append(random.randint(1,100))

print("List Elements are:",l)

avg=sum(l)/len(l)

print("The Average of the Elements in the list is=",avg)

print("Largest Value in the list is :",max(l))

print("Smallest Value in the list is :",min(l))

l1=sorted(l)

print("second Largest value in the List is:",l1[-2])

print("Second smallest value in the List is:",l1[1])

count=0

for i in l1:

    if (i%2==0):

        count=count+1

print("Number of Even Number in the List is :",count)
```

Output:

List Elements are: [23, 72, 37, 86, 9, 54, 9, 78, 32, 40, 12, 3, 54, 65, 49, 88, 98, 70, 51, 82]

The Average of the Elements in the list is= 50.6

Largest Value in the list is : 98

Smallest Value in the list is : 3

second Largest value in the List is: 88

Second smallest value in the List is: 9

Number of Even Number in the List is : 12

List Elements are: [31, 56, 77, 32, 47, 85, 17, 61, 86, 50, 86, 85, 96, 14, 37, 64, 75, 11, 22, 68]

The Average of the Elements in the list is= 55.0

Largest Value in the list is : 96

Smallest Value in the list is : 11

second Largest value in the List is: 86

Second smallest value in the List is: 14

Number of Even Number in the List is : 10

Program-13

Write a program that asks the user for an integer and creates a list that consists of the factors of that integer.

Program:

```
num=int(input("Enter the Number:"))
```

```
l=[]
```

```
for i in range(1,num+1):
```

```
    if(num%i==0):
```

```
        l.append(i)
```

```
print("factors of %d is"%(num),l)
```

Output:

Enter the Number:45

factors of 45 is= [1, 3, 5, 9, 15, 45]

Enter the Number:16

factors of 16 is= [1, 2, 4, 8, 16]

Program -14

Write a program that generates 100 random integers that are either 0 or 1. Then find the longest run of zeros, the largest number of zeros in a row. For instance, the longest run of zeros in [1,0,1,1,0,0,0,0,1,0,0] is 4.

```
Import random
```

```
l=[]
```

```
for num in range(20):
```

```
    l.append(random.randint(0,1))
```

```
print("ListElementsare:",l)
```

```
c=0
```

```
max_count=0
```

```
for range(0,20):
```

```
    if (l[i]== 0):
```

```
        c += 1
```

```
        if(i==len(l)-1):
```

```
            if(c>max_count):
```

```
                max_count=c
```

```
    else:
```

```
        if c > max_count:
```

```
            max_count = c
```

```
        c = 0
```

```
print("Longest Run of Zeros in a row is:",max_count)
```

OUTPUT:

List Elements are: [0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0]

Longest Run of Zeros in a row is: 2

List Elements are: [0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0]

Longest Run of Zeros in a row is: 3

Program-15

Write a program that removes any repeated items from a list so that each item appears at most once. For instance, the list [1,1,2,3,4,3,0,0] would become [1,2,3,4,0].

Program:

```
n=int(input("Enter the number elements in the list:"))
l=[]
for i in range(1,n+1):
    l.append(int(input("Enter the list %d Element:"%(i))))

print("The List elements are",l)
r=[]
for i in l:
    if i not in r:
        r.append(i)
print("remove repeated items from a list is:",r )
```

Output:

```
Enter the number elements in the list:5
Enter the list 1 Element:45
Enter the list 2 Element:1
Enter the list 3 Element:2
Enter the list 4 Element:5
Enter the list 5 Element:45
The List elements are [45, 1, 2, 5, 45]
remove repeated items from a list is: [45, 1, 2, 5]
```

```
Enter the number elements in the list:7
```

```
Enter the list 1 Element:1
```

```
Enter the list 2 Element:1
```

```
Enter the list 3 Element:2
```

```
Enter the list 4 Element:5
```

```
Enter the list 5 Element:6
```

```
Enter the list 7 Element:3
```

```
The List elements are [1, 1, 2, 5, 6, 2, 3]
```

```
remove repeated items from a list is: [1, 2, 5, 6, 3]
```

Program 16

Write a program that asks the user to enter a length in feet. The program should then give the user the option to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometers. Say if the user enters a 1, then the program converts to inches, if they enter a 2, then the program converts to yards, etc. While this can be done with if statements, it is much shorter with lists and it is also easier to add new conversions if you use lists.

Program:

```
feet=int(input("Enter the Length in Feet:"))
print("""Choose 1 to convert into inches,
choose 2 to convert into yards,
choose 3 to convert into miles,
choose 4 to convert into millimeters,
choose 5 to convert into centimeters,
choose 6 to convert into meters,
choose 7 to convert into kilometers""")

opt=int(input('Enter the your choice:'))

if(opt==1):
    print("Inches is:",feet * 12)
elif(opt==2):
    print("Yards is:",feet * 0.33333)
elif(opt==3):
    print("Miles is:",feet * 0.000189393939)
elif(opt==4):
    print("MilliMeters is:",feet * 304.8)
elif(opt==5):
    print("centimeters is:",feet * 30.48)
elif(opt==6):
    print("meters is:",feet * 0.3048)
elif(opt==7):
    print("kilometersis:",feet * 0.0003048)
```

Output:

Enter the Length in Feet:6
Choose 1 to convert into inches,
choose 2 to convert into yards,
choose 3 to convert into miles,
choose 4 to convert into millimeters,
choose 5 to convert into centimeters,
choose 6 to convert into meters,
choose 7 to convert into kilometers
Enter the your choice:1
Inches is: 72

Enter the Length in Feet:6
Choose 1 to convert into inches,
choose 2 to convert into yards,
choose 3 to convert into miles,
choose 4 to convert into millimeters,
choose 5 to convert into centimeters,
choose 6 to convert into meters,
choose 7 to convert into kilometers
Enter the your choice:2
Yards is: 1.99998

Program 17

Write a function called sum_digits that is given an integer num and returns the sum of the digits of num.

Program:

```
def sum_digit(num):  
    sum1=0  
    while(num!=0):  
        sum1=sum1+num%10  
        num=num//10  
    return(sum1)  
num=int(input("enter the integer number:"))  
sum1=sum_digit(num)  
print("the sum of %d digits are =%d"%(num,sum1))
```

Output:

enter the integer number:164
the sum of 164 digits are =11

enter the integer number:124
the sum of 124 digits are =7

Program 18

Write a function called *first_diff* that is given two strings and returns the first location in which the strings differ. If the strings are identical, it should return -1.

Program:

```
def first_diff(s1,s2):  
    if(s1==s2):  
        return(-1)  
    else:  
        if len(s1)==len(s2):  
            for i in range(len(s2)):  
                if s1[i]!=s2[i]:  
                    return(i+1)  
s1=input("Enter the string1:")  
s2=input("Enter the string2:")  
c=first_diff(s1,s2)  
if(c==-1):  
    print("Both Strings are Identical")  
else:  
    print("first Difference occurs at location: ",c)
```

Output:

Enter the string1:crr

Enter the string2:crr

Both Strings are Identical

Enter the string1:crrcoe

Enter the string2:crrcse

first Difference occurs at location: 5

Program 19

Write a function called number_of_factors that takes an integer and returns how many factors the number has.

Program:

```
def number_of_factors(num):  
    count=0  
    for i in range(1,num+1):  
        if(num%i==0):  
            print(i,end=" ")  
            count=count+1  
    return(count)
```

```
num=int(input("Enter the integer:"))  
c=number_of_factors(num)  
print("\n factors the %d number has =%d"%(num,c))
```

Output:

Enter the integer:25

1 5 25

factors the 25 number has =3

Enter the integer:16

1 2 4 8 16

factors the 16 number has =5

Program 20

Write a function called is sorted that is given a list and returns True if the list is sorted and False otherwise.

Program:

```
def is_sorted(l):  
    k=l[:]  
    k.sort()
```

```

print("The K List elements are",k)
if(l==k):
    return True
else:
    return False
n=int(input("Enter the number elements in the list:"))
l=[]
for i in range(1,n+1):
    l.append(int(input("Enter the list %d Element:"%(i))))
print("The L List elements are",l)
Print(is_sorted(l))

```

Output:

```

Enter the number elements in the list:5
Enter the list 1 Element:1
Enter the list 2 Element:2
Enter the list 3 Element:3
Enter the list 4 Element:4
Enter the list 5 Element:5
The L List elements are [1, 2, 3, 4, 5]
The K List elements are [1, 2, 3, 4, 5]
True

```

```

Enter the number elements in the list:5
Enter the list 1 Element:3
Enter the list 2 Element:2
Enter the list 3 Element:1
Enter the list 4 Element:5
Enter the list 5 Element:6
The L List elements are [3, 2, 1, 5, 6]
The K List elements are [1, 2, 3, 5, 6]
False

```

Program 21

Write a function called root that is given a number x and an integer n and returns $x^{1/n}$. In the function definition, set the default value of n to 2.

Program:

```

def root(x,n=2):
    s=(x**(1/n))
    return(s)

```

```

x=int(input("enter the X value:"))

```

```

n=int(input("enter the nvalue:"))
res1=root(x)
res2=root(x,n)
print("Root value With Default 'n' value 2 is=",res1)
print("Root value With 'n' value is=",res2)

```

Output:

```

enter the X value:18
enter the nvalue:2
Root value With Default 'n' value 2 is= 4.242640687119285
Root value With 'n' value is= 4.242640687119285

```

```

enter the X value:12
enter the nvalue:4
Root value With Default 'n' value 2 is= 3.4641016151377544
Root value With 'n' value is= 1.8612097182041991

```

```

enter the X value:16
enter the n value:4
Root value With Default 'n' value 2 is= 4.0
Root value With 'n' value is= 2.0

```

PROGRAM-22

Write a function called primes that is given a number n and returns a list of the first n primes. Let the default value of n be 100.

Program:

```

def
printPrime(n=100):
    l=[]
    x=2
    while(len(l)<n):
        for i in range(2,int(x**0.5)+1):
            if(x%i==0):
                break
            else:
                l.append(x)
                x=x+1
        return(l)
n=int(input("enter the number of prime numbers:"))
s=printPrime(n)
print("List of first",n,"primes:",s)
j=printPrime()
print("List of first 100 primes:",j)

```


Output:

enter the number of prime numbers:10

List of first 10 primes: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]

List of first 100 primes: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541]

Program 23

Write a function called merge that takes two already sorted lists of possibly different lengths, and merges them into a single sorted list.

(a) Do this using the sort method. (b) Do this without using the sort method.

```
def merge(l1,l2):
```

```
    l=l1+l2
```

```
    l.sort()
```

```
    return l
```

```
l1=list(map(int,input("Enter the Sorted list 1:").split()))
```

```
l2=list(map(int,input("Enter the Sorted list 2:").split()))
```

```
s=merge(l1,l2)
```

```
print("After merge the list is:",s)
```

Output:

Enter the Sorted list 1:1 4 6 8 9

Enter the Sorted list 2:2 6 8 9 90

After merge the list is: [1, 2, 4, 6, 6, 8, 8, 9, 9, 90]

without using the sort method

```
def merge_lists(L1, L2):
```

```
    # When one of them is an empty list, returns the other list
```

```
    if not L1:
```

```
        return L2
```

```

elif not L2:
    return L1

result = []
i = 0
j = 0

for k in range(len(L1) + len(L2)):
    if L1[i] <= L2[j]:
        result.append(L1[i])
        if i < len(L1) - 1:
            i += 1
        else:
            result += L2[j:] # When the last element in L1 is reached,
            break          # append the rest of L2 to result.
    else:
        result.append(L2[j])
        if j < len(L2) - 1:
            j += 1
        else:
            result += L1[i:] # When the last element in L2 is reached,
            break          # append the rest of L1 to result.

return result
l1=list(map(int,input("Enter the Sorted list 1:").split()))
l2=list(map(int,input("Enter the Sorted list 2:").split()))
s=merge_lists(l1,l2)
print("After merge the list is:",s)

```

Output:

```

Enter the Sorted list 1:2 3 4 78
Enter the Sorted list 2:1 89 23 90
After merge the list is: [1, 2, 3, 4, 78, 89, 23, 90]

```

Program 24

Write a program that asks the user for a word and finds all the smaller words that can be made from the letters of that word. The number of occurrences of a letter in a smaller word can't exceed the number of occurrences of the letter in the user's word.

Program:

```

from itertools import permutations
w=input("Enter A Word:")
for i in range(2,len(w)):

```

```
for p in permutations(w,i):
    print(" ".join(p),end=' ')
```

Output:

```
Enter A Word: CSE
CS CE SC SE EC ES
```

```
Enter A Word:crre
```

```
cr cr ce rc rr re rc rr re ec er er crr cre crr cre cer cer rcr rce rrc rre rec rer rcr rce rrc rre rec rer ecr
ecr erc err erc err
```

Program 25

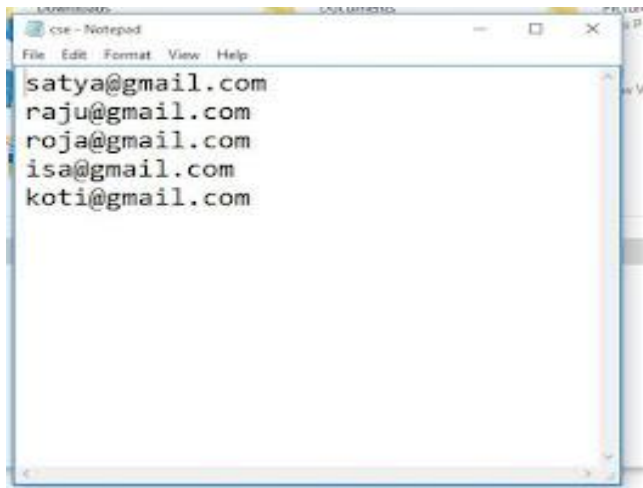
Write a program that reads a file consisting of email addresses, each on its own line. Your program should print out a string consisting of those email addresses separated by semicolons.

Program:

```
file=open(input("Enter the File name:"))
l=file.readlines()
for line in range(len(l)):
    if (l==len(l)-1):
        print('{}'.format(l[line].strip()))
    else:
        print('{}'.format(l[line].strip()),end=";")
```

Output:

```
Enter the File name:cse.txt
satya@gmail.com;raju@gmail.com;roja@gmail.com;isa@gmail.com;koti@gmail.com;
```



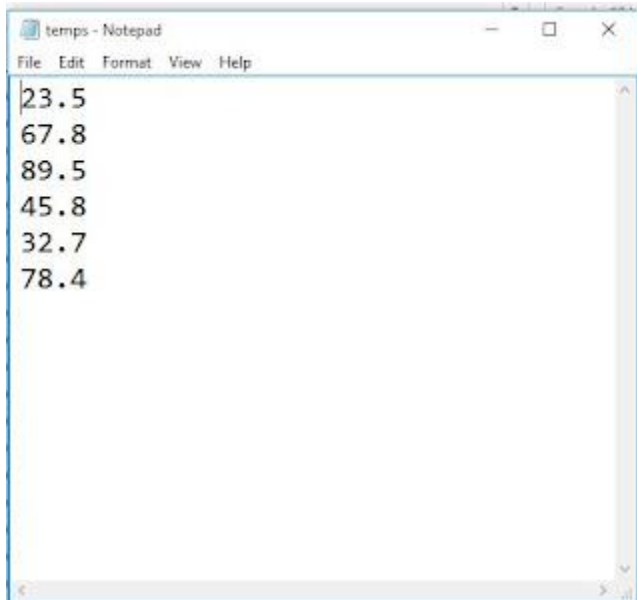
Program 26

Write a program that reads a list of temperatures from a file called *temps.txt*, converts those temperatures to Fahrenheit, and writes the results to a file called *ftemps.txt*.

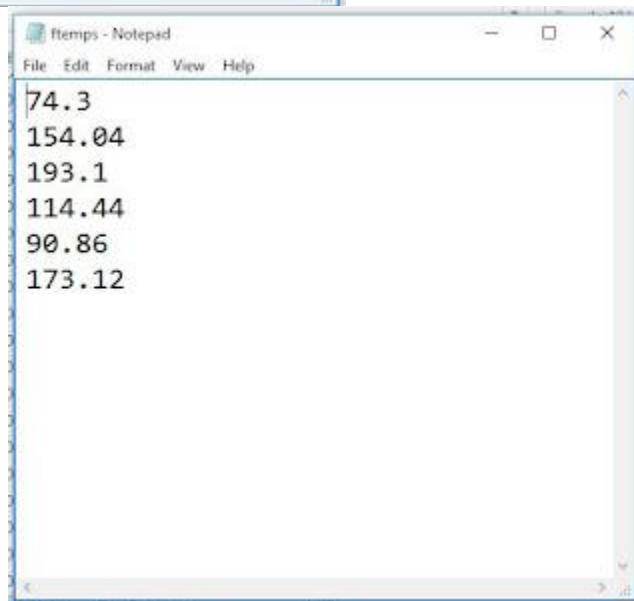
Program:

```
file=open('temps.txt','r')
l=file.readlines()
f=open('ftemps.txt','w')
for line in range(len(l)):
    d=l[line].strip()
    fahrenheit = (float(d) * 9/5) + 32
    f.write(str(fahrenheit)+"\n")
f.close()
```

Output



```
temps - Notepad
File Edit Format View Help
23.5
67.8
89.5
45.8
32.7
78.4
```



```
ftemps - Notepad
File Edit Format View Help
74.3
154.04
193.1
114.44
90.86
173.12
```

Program 27

Write a class called `Product`. The class should have fields called `name`, `amount`, and `price`, holding the product's name, the number of items of that product in stock, and the regular price of the product. There should be a method `get_price` that receives the number of items to be bought and returns a the cost of buying that many items, where the regular price is charged for orders of less than 10 items, a 10% discount is applied for orders of between 10 and 99 items, and a 20% discount is applied for orders of 100 or more items. There should also be a method

called *make_purchase* that receives the number of items to be bought and decreases amount by that much.

Program:

```
class Product:

    def __init__(self, name, total_items, price):

        self.name = name

        self.total_items = total_items

        self.price = price

    def get_price(self, number_to_be_bought):

        discount = 0

        if number_to_be_bought < 10:

            print("Regular price is charged for your order")

            print("NO Discount\n ")

            cost=self.price * number_to_be_bought

            print('Final costs = ',cost)

            print(" A 10% discount is applied for orders of between 10 and 99 items")

            print(" A 20% discount is applied for orders of 100 or more items")

        elif 10 <= number_to_be_bought < 99:

            Actual_cost=self.price * number_to_be_bought

            print('Actual cost is = ',Actual_cost)

            discount = 10

            p = (100 - discount) / 100 * self.price
```

```

cost=p * number_to_be_bought

print('Final costs(Discount cost10%) is = ',cost)

print("you save=",Actual_cost-cost,"Rupes")

else:

    Actual_cost=self.price * number_to_be_bought

    print('Actual cost is = ',Actual_cost)

    discount = 20

    p = (100 - discount) / 100 * self.price

    cost=p * number_to_be_bought

    print('Final costs(Discount cost20%) = ',cost,)

    print("you save=",Actual_cost-cost,"Rupes")

name=input("Enter the Name of the product:\n")

total_items=int(input("Total Number of items:\n")),

price=int(input('Digit price of each item:\n'))

p=Product(name,total_items,price)

number_to_be_bought=int(input("Enter Number of items you want u buy:"))

p.get_price(number_to_be_bought)

```

Output:

```

Enter the Name of the product:
pen
Total Number of items:
200
Digit price of each item:

```

5

Enter Number of items you want u buy:7

Regular price is charged for your order

NO Discount

Final costs = 35

A 10% discount is applied for orders of between 10 and 99 items

A 20% discount is applied for orders of 100 or more items

Program 28

Write a class called Time whose only field is a time in seconds. It should have a method called *convert_to_minutes* that returns a string of minutes and seconds formatted as in the following example: if seconds is 230, the method should return '5:50'. It should also have a method called *convert_to_hours* that returns a string of hours, minutes, and seconds formatted analogously to the previous method.

Program:

```
class Time:

    def __init__(self,sec):

        self.sec=sec

    def seconds_to_Minutes(self):

        s=self.sec

        minutes = s // 60

        seconds=s%60
```



```

        print("Times in Minutes:seconds format",str(minutes)+":",str(seconds))

def seconds_to_Hours(self):

    s=self.sec

    hours = s // 3600

    minutes=(s//60)% 60

    seconds=s%60

print("TimesinHours:Minutes:seconds format",str(hours)+":",str(minutes)+":",str(seconds))

sec=int(input('Enter the number of seconds:'))

obj=Time(sec)

obj.seconds_to_Minutes()

obj.seconds_to_Hours()

```

Output:

Enter the number of seconds:150

Times in Minutes:seconds format 2: 30

Times in Hours:Minutes:seconds format 0: 2: 30

Enter the number of seconds:4500

Times in Minutes:seconds format 75: 0

Times in Hours:Minutes:seconds format 1: 15: 0

Program 29

Write a class called Converter. The user will pass a length and a unit when declaring an object from the class—for example, `c = Converter(9,'inches')`. The possible units are inches, feet, yards, miles, kilometers, meters, centimeters, and millimeters. For each of these units there should be a method that returns the length converted into those units. For example, using the Converter object created above, the user could call `c.feet()` and should get 0.75 as the result.

Program:

Class converter:

```
def __init__(self,length,unit) :  
    self.length=length  
    self.unit=unit  
def feet(self):  
    if (self.unit=='feet')  
        return self.length  
    elif (self.unit=='inches')
```

```
        return self.length/12
    elif (self.unit=='yards')
        return self.length/0.333
    elif (self.unit=='miles')
        return self.length/0.000189
    elif (self.unit=='millimeters')
        return self.length/304.8
    elif (self.unit=='centimeters')
        return self.length/30.48
    elif (self.unit=='meters')
        return self.length/0.305
    elif (self.unit=='kilometers')
        return self.length/0.000305
```

```
def inches (self):
```

```
    if (self.unit=='feet')
        return self.length*12
    elif (self.unit=='inches')
        return self.length
    elif (self.unit=='yards')
        return self.length*36
    elif (self.unit=='miles')
        return self.length*63360
    elif (self.unit=='millimeters')
        return self.length*0.0393701
```

```
elif (self.unit=='centimeters')
    return self.length*0.393701
elif (self.unit=='meters')
    return self.length*39.3701
elif (self.unit=='kilometers')
    return self.length*39370.1
```

```
def yards(self):
```

```
    if (self.unit=='feet')
        return self.length*0.333333
    elif (self.unit=='inches')
        return self.length*0.0277778
    elif (self.unit=='yards')
        return self.length
    elif (self.unit=='miles')
        return self.length*1760
    elif (self.unit=='millimeters')
        return self.length*0.00109361
    elif (self.unit=='centimeters')
        return self.length*0.0109361
    elif (self.unit=='meters')
        return self.length*1.09361
    elif (self.unit=='kilometers')
        return self.length*109.361
```

```
def miles(self):
    if (self.unit=='feet')
        return self.length*0.000189394
    elif (self.unit=='inches')
        return self.length*63360
    elif (self.unit=='yards')
        return self.length*0.027777728
    elif (self.unit=='miles')
        return self.length
    elif (self.unit=='millimeters')
        return self.length/1609344
    elif (self.unit=='centimeters')
        return self.length/160934.4
    elif (self.unit=='meters')
        return self.length/1609.344
    elif (self.unit=='kilometers')
        return self.length*/1.609
```

```
def kilometers(self):
    if (self.unit=='feet')
        return self.length/3280.84
    elif (self.unit=='inches')
        return self.length/39370.1
    elif (self.unit=='yards')
        return self.length/1093.61
```

```

elif (self.unit=='miles')
    return self.length/0.621371
elif (self.unit=='millimeters')
    return self.length/1000000
elif (self.unit=='centimeters')
    return self.length/100000
elif (self.unit=='meters')
    return self.length/1000
elif (self.unit=='kilometers')
    return self.length
def meters(self):
    if (self.unit=='feet')
        return self.length/3.28084
    elif (self.unit=='inches')
        return self.length/39.3701
    elif (self.unit=='yards')
        return self.length/1.09361
    elif (self.unit=='miles')
        return self.length/0.000621371
    elif (self.unit=='millimeters')
        return self.length/1000
    elif (self.unit=='centimeters')
        return self.length/100
    elif (self.unit=='meters')
        return self.length

```

```

elif (self.unit=='kilometers')
    return self.length/0.001
def centimeters(self):
    if (self.unit=='feet')
        return self.length/0.0328084
    elif (self.unit=='inches')
        return self.length/0.393701
    elif (self.unit=='yards')
        return self.length/0.0109361
    elif (self.unit=='miles')
        return self.length*160934
    elif (self.unit=='millimeters')
        return self.length/10
    elif (self.unit=='centimeters')
        return self.length
    elif (self.unit=='meters')
        return self.length*100
    elif (self.unit=='kilometers')
        return self.length*100000

def millimeters(self):
    if (self.unit=='feet')
        return self.length*304.8
    elif (self.unit=='inches')
        return self.length*0.0393701

```

```

elif (self.unit=='yards')
    return self.length*0.00109361
elif (self.unit=='miles')
    return self.length*1609340
elif (self.unit=='millimeters')
    return self.length
elif (self.unit=='centimeters')
    return self.length*10
elif (self.unit=='meters')
    return self.length*100
elif (self.unit=='kilometers')
    return self.length*1000000

```

```
len=int(input("Enter length"))
```

```
type=input("Enter unit type: inches,feet,yards,miles,millimeters,centimeters,meters,kilometers---
->")
```

```
c=Converter(len,type)
```

```
print("Length in Feet: ",round(c.feet(),3))
```

```
print("Length in Inches: ",round(c.inches(),3))
```

```
print("Length in Yards: ",round(c.yards(),3))
```

```
print("Length in Miles: ",round(c.miles(),3))
```

```
print("Length in Kilometers: ",round(c.kilometers(),3))
```

```
print("Length in Meters: ",round(c.meters(),3))
```

```
print("Length in Centimeters: ",round(c.centimeters(),3))
```

```
print("Length in Millimeters: ",round(c.millimeters(),3))
```


Output

Enter length: 6

Enter unit type: inches,feet,yards,miles,millimeters,centimeters,meters,kilometers---->inches

Length in Feet :0.5

Length in Inches : 6

Length in Yards: 0.167

Length in Miles: 380160

Length in Kilometers:0

Length in Meters: 0.152

Length in Centimeters:15.24

Length in Millimeters:152.4

Program 30

Write a Python class to implement pow(x, n).

Program:

```
class Pow1:

    def poe(self,x,n):

        k=x**n

        print("pow({},{})={}".format(x,n,k))

p=Pow1()

x=int(input("Enter the X value:"))

n=int(input("Enter the Y value:"))

p.poe(x,n)
```

Output:

Enter the X value:3
Enter the Y value:4
pow(3,4)=81

Program 31

Write a Python class to reverse a string word by word.

Program:

```
class Sa:

    def display(self,s):

        words = s.split(' ')

        reverse_sentence = ''.join(reversed(words))

        print("The reverse words of string:",reverse_sentence)

k=Sa()

s = input("enter the string:")

k.display(s)
```

Output:

enter the string:welcome to python

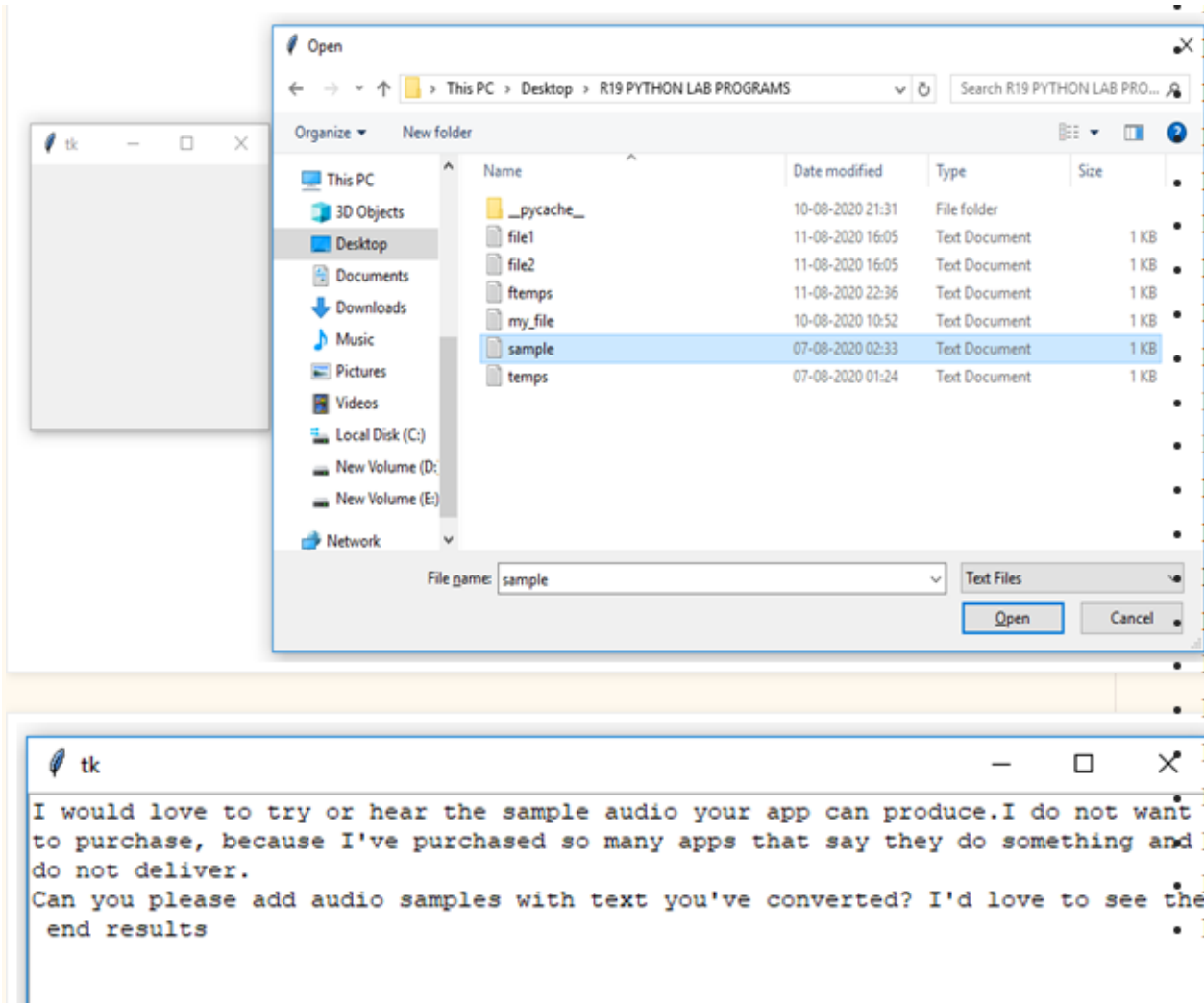
The reverse words of string: python to welcome

Program-32

Write a program that opens a file dialog that allows you to select a text file. The program then displays the contents of the file in a textbox.

```
from tkinter import filedialog
from tkinter import Tk
from tkinter import *
root=Tk()
root.filename=filedialog.askopenfilename(filetypes=(("Text Files", ".txt"),("All Files", " *")))
text1=open(root.filename).read()
T=Text(root,height=25,width=80)
T.pack()
T.insert(END,text1)
root.mainloop()
```

Output



Program 33

Write a program to demonstrate Try/except/else.

Program:

```
try:
    a=int(input("enter the A value:"))
    b=int(input("enter the B value:"))
    s=a/b
except ZeroDivisionError:
    print("Can't divide by zero(b=0)")
else:
    print("The division is=",s)
```

Output:

enter the A value:27
enter the B value:3
The division is= 9.0

enter the A value:70
enter the B value:0
Can't divide by zero(b=0)

Program 34

Write a program to demonstrate try/finally and with/as.

Program:

```
try:  
    a=int(input("enter the A value:"))  
    b=int(input("enter the B value:"))  
    s=a/b  
except ZeroDivisionError:  
    print("Can't divide by zero(b=0)")  
else:  
    print("The division is=",s)  
finally:  
    print("This is always executed")
```

Output:

enter the A value:20
enter the B value:0
Can't divide by zero(b=0)
This is always executed

enter the A value:20

enter the B value:2

The division is= 10.0

This is always executed

CONTENT BEYOND THE SYLLABUS

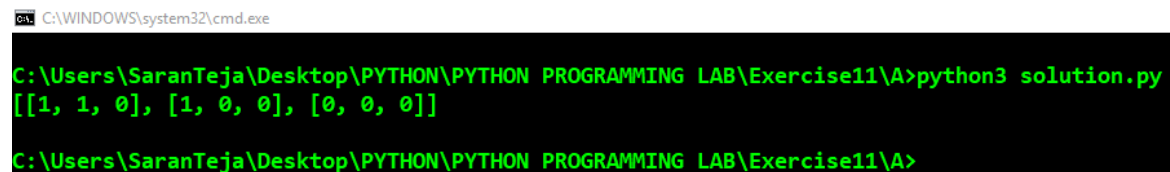
1. Write a program that defines a matrix and prints

```
# Write a program that defines a matrix and prints

def main():
    matrix = [
        [1, 1, 0],
        [1, 0, 0],
        [0, 0, 0]
    ]
    print(matrix)

if __name__ == '__main__':
    main()
```

OUTPUT:



```
C:\WINDOWS\system32\cmd.exe
C:\Users\SaranTeja\Desktop\PYTHON\PYTHON PROGRAMMING LAB\Exercise11\A>python3 solution.py
[[1, 1, 0], [1, 0, 0], [0, 0, 0]]
C:\Users\SaranTeja\Desktop\PYTHON\PYTHON PROGRAMMING LAB\Exercise11\A>
```

2. Write a program to perform addition of two square matrices

```

# Write a program to perform addition of two square matrices

def matrix_addition(X, Y):
    return [[X[i][j] + Y[i][j] for j in range(len(X[0]))] for i in range(len(X))]

def main():

    n = int(input("Enter n value\n"))
    m = int(input("Enter m value\n"))

    if n != m:
        print('n and m length should be same')
        return

    X, Y = [], []

    print('Enter n matrix values')
    for i in range(n):
        X.append([int(x) for x in input().split()])
    print('Enter m matrix values')
    for i in range(m):
        Y.append([int(x) for x in input().split()])

    print(matrix_addition(X, Y))

if __name__ == '__main__':
    main()

```

OUTPUT:

C:\WINDOWS\system32\cmd.exe

```

C:\Users\SaranTeja\Desktop\PYTHON\PYTHON PROGRAMMING LAB\Exercise11\B>python3 solution.py
Enter n value
3
Enter m value
3
Enter n matrix values
3
3
2
Enter m matrix values
4
5
8
[[7], [8], [10]]

C:\Users\SaranTeja\Desktop\PYTHON\PYTHON PROGRAMMING LAB\Exercise11\B>

```